



Conceptos básicos de UML

Ing. Luis Guillermo Molero Suárez



UNIFIED
MODELING
LANGUAGE™

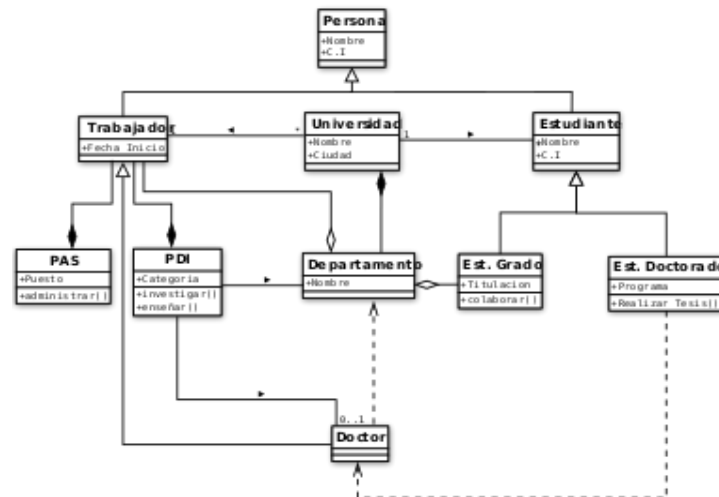


¿Qué es un modelo?

Un modelo se define como una representación que toma los elementos más significativos de lo que se trata de modelar y simplifica/excluye el resto; muchos aspectos de la ingeniería hacen uso de modelos.

Por otra parte, un modelo de un sistema está edificado con un lenguaje de modelado que tiene semántica (sentido), notación (sistemas de signos y/o símbolos) y ser representados a través de textos y/o gráficos. (Rumbaugh, Jacobson, & Booch, 2000).

Diagrama de Clases



En ese sentido y según Rumbaugh et al (2000), los modelos se usan para muchos propósitos, entre ellos:

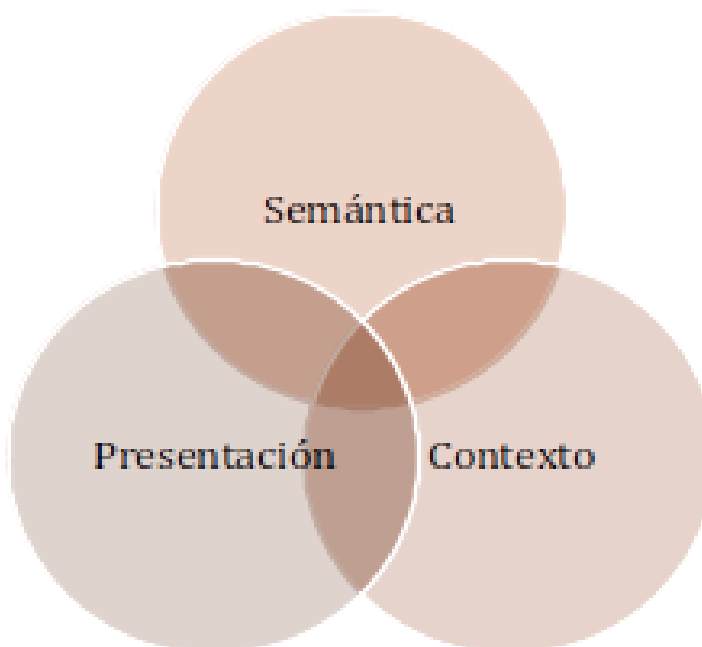


- Para obtener y especificar claramente los requerimientos y el dominio de conocimiento, a los fines de que los interesados del sistemas estén de acuerdo con ellos.
- Para reflexionar acerca del modelo de un sistema.
- Para decidir acerca de especificaciones del diseño de un sistema considerando sus requisitos funcionales.
- Para concebir productos que generen valor al negocio.
- Para optimizar la información en magnos sistemas de información.
- Para discernir sobre los aspectos económicos de varias soluciones.
- Para reducir a la simplicidad sistemas enmarañados.

Otro aspecto importante para resaltar de los modelos es que ellos adquieren diferentes configuraciones de acuerdo su propósito, por lo cual, los niveles de detalle del modelo deben adecuarse a las siguientes necesidades:

- Orientar al proceso de pensamiento.
- Descripción genérica de la organización principal de un sistema.
- Descripción holística de un sistema de información final.
- Muestras de sistemas distintivos o alcanzables.
- Representaciones finales o parciales de un sistema.

En la siguiente figura, se ilustran los elementos que define un modelo





¿Qué es UML?

El Lenguaje de Modelado Unificado (UML) nació a finales de los 80's y amalgama los métodos de Booch, Jacobson y Rumbaugh para alcanzar un dominio no obstante más extenso, asimismo, se establece que UML representa un lenguaje de modelado más no un método; este representa una notación gráfica de lo cual se corresponden los métodos para representar los diseños, es decir, el punto de partida del cual hacen uso los métodos para la fase de diseño de software. (Rumbaugh, Jacobson, & Booch, 2000)



Uno de los elementos más importantes de UML es que corresponden a un lenguaje o notación estructurada que delinea los conceptos de software en forma de diagrama, que viene de la combinación de una gran cantidad de estándares de la comunidad informática mundial y permite establecer los procedimientos para el diseño de software antes de su construcción. (Martin, 2004).

De acuerdo con Rumbaugh et al (2000), algunas de las propiedades de UML son:

- Es un lenguaje distribuido ajustado a las necesidades de conectividad
- Reconocido ampliamente por la industria informática mundial.
- Como una mixtura de múltiples notaciones, tiene la propiedad de reemplazar a casi todas ellas.
- Puede modelar estructuras complejas.
- Soporta fundamentalmente estructuras orientadas a objeto.
- Su notación es esencialmente abstracta.

Finalmente, UML es considerado un lenguaje de ámbito general para el modelamiento en diseños orientados a objetos, que acuerda una sintaxis resultante a partir de: Un modelo orientado a objetos, modelo de datos, modelado de componentes y modelos flujos de trabajo. UML, no simboliza un lenguaje de programación, por consiguiente, es manipulado como instrumento en diversos lenguajes de programación orientados a objetos. (Larman, 2003).

UML en el diseño orientado a objeto

El estudio de algoritmos se define en la forma de incorporar paradigmas o modelos de datos, en ese sentido, existen diversas categorías de modelos para resolver problemas entre los cuales, los lenguajes orientados a objetos que se expresan precisando objetos y que estos se correspondan entre sí. Por su parte



los objetos, son elementos que deben ser operados y tienen presencia en el universo real, también, tienen caracteres que los diferencian entre ellos (UML y su función en el modelado y diseño orientados a objetos, s.f.).

Los lenguajes de programación orientada a objetos (Lenguaje OO) inundan el mercado actual en la materia, esto por su propiedad de manipular los objetos en el desarrollo de sistemas tal y como lo hace UML. En la siguiente figura, se ilustran los elementos que opera UML.



En suma, las metodologías de UML fueron diseñadas especialmente más no únicamente para ayudar a los programadores a elaborar un buen desarrollo orientado a objetos. (Forwel & Scott, 1997).

UML y el modelado de datos

Un lenguaje de modelado establece la forma de expresar los distintos modelos que se pueden dar lugar en el proceso de desarrollo de software, asimismo, define un grupo de elementos del modelo y se centra en los diagramas sin pasar por alto el texto. (Stevens & Pooley, 2002)

Ejemplo: UML corresponde a un lenguaje de modelado.

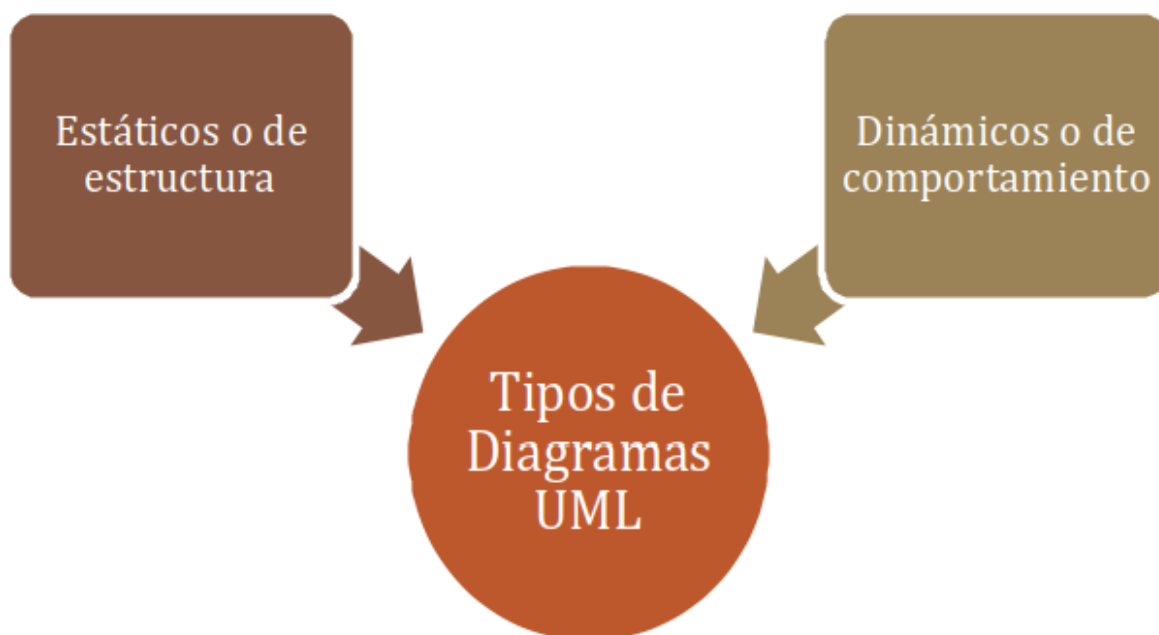
En ese sentido, un lenguaje de modelado define un conjunto de símbolos estandarizados y el modo de cómo disponer de cada uno para modelar un diseño de software orientado a objetos y frecuentemente son usados en conjunto con

metodologías de desarrollo de software establecer un plan de implementación que sirve a su vez para informar al equipo de desarrollo.

Tipos de diagramas UML

Existen dos tipos de diagramas UML:

- Estáticos o de estructura
- Dinámicos o de comportamiento




Cuando nos referimos a los diagramas de tipo estáticos o de estructura, hablamos de diagramas que son utilizados en diferentes momentos del proyecto y con distintos objetos, solo para precisar la organización estática de un sistema y sus partes en diferentes niveles de abstracción. Asimismo, dependiendo de la dimensión del proyecto se pueden exhibir uno o varios diagramas con distintas vistas.

Además, estos diagramas abrigan elementos afines a los momentos iniciales del ciclo de vida de un proyecto y permiten relacionar desde los requerimientos funcionales, arquitectura, diseño de módulos hasta la configuración del hardware donde serán desplegados. (Pantaleo & Rinaudo, 2015).

Por otra parte, cuando nos referimos a los diagramas de tipo dinámicos o de comportamiento, existe la similitud con los diagramas estáticos cuando hablamos de que son utilizados en diferentes momentos del proyecto y con distintos objetos,

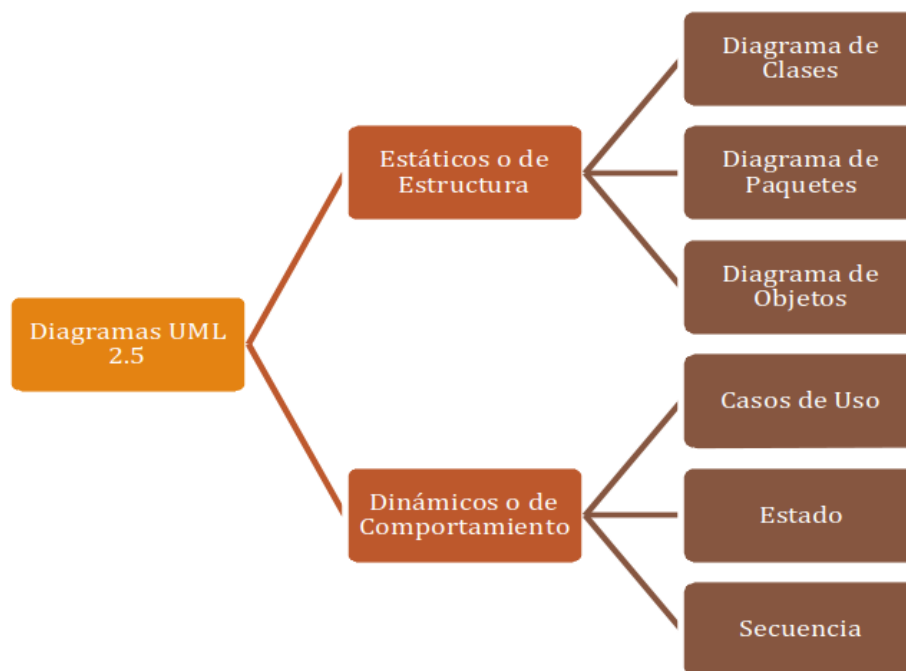


sin embargo, precisan la organización dinámica de un sistema, es decir, las vistas dinámicas. Asimismo, dependiendo de la dimensión del proyecto se pueden exhibir uno o varios diagramas con distintas vistas.

	Los diagramas de tipo estáticos, precisan la organización estática de un sistema, mientras que los diagramas de tipo dinámicos precisan la organización dinámica de un sistema
---	--



En contraste con los diagramas estáticos, los de comportamiento muestran la actuación de un sistema de forma dinámica, es decir, muestran los cambios que desarrolla el sistema mientras se encuentra en ejecución.

En la siguiente imagen, se exponen los diagramas UML de tipo estáticos y dinámicos.





Enlaces recomendados:

	<p>Los siguientes recursos, sirven para consolidar conocimientos sobre la introducción a UML</p> <ul style="list-style-type: none">• https://www.youtube.com/watch?v=-OWd0tJAK10• https://www.youtube.com/watch?v=3LrOpwhsplY&t=2s• https://www.youtube.com/watch?v=qoxoX3ZiazI&t=1s
	<p>Para más información acerca de los diagramas de clases de UML, puedes consultar:</p> <ul style="list-style-type: none">• Introducción al UML: Lenguaje para modelar objetos• Sistemas organizacionales. Teoría y práctica <p>Disponibles en: https://books.google.es/</p>